Digital Image Processing
Lab Assignment

2022-23

Department of Computer Science & Applications
Institute of Engineering & Technology

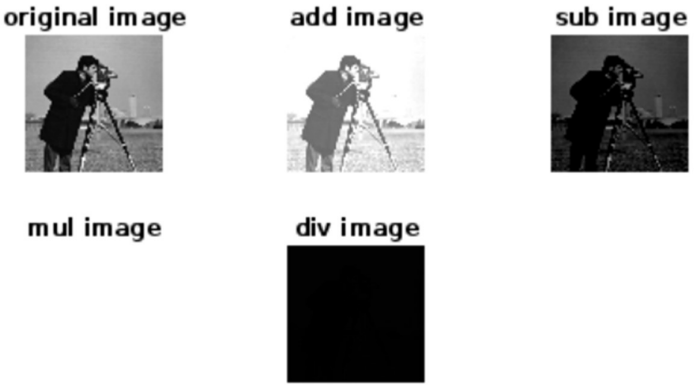
Submitted to:
Mr. Pradeep Kumar Singh

Submitted By:
Anubhav Varshney
Btech CS(AIML)
3rd Year(201550029)

Assignment 1

% Create command to familiarize with MATLAB & Create the matrices & perform the various operations on them.

```
a=[1,2,3,4,5;6,7,8,9,0,];
[r,c]=size(a);
a1=zeros(4,2);
a2=ones(3,4);
A = [10 20 30; 11 12 13; 40 50 60]; B = [51 52 53; 21 23 21; 44 54 64];
addition = A+B;
subtraction = AB;
multiplication = A*B;
division = A/B;
l = length(a);
diagonal_ele = diag(A);
identity_mat = eye(3,3);
sin_value=sin(A);
T=transpose(A);
S = sum(T);
inverse_mat = inv(A);
detminant_mat = det(A);
rank_mat = rank(A);
[V, D] = eig(A);
[2+3i 1+9i; 6+7i 9+i];
c1 = abs(C);
R = rand (3,4)
```

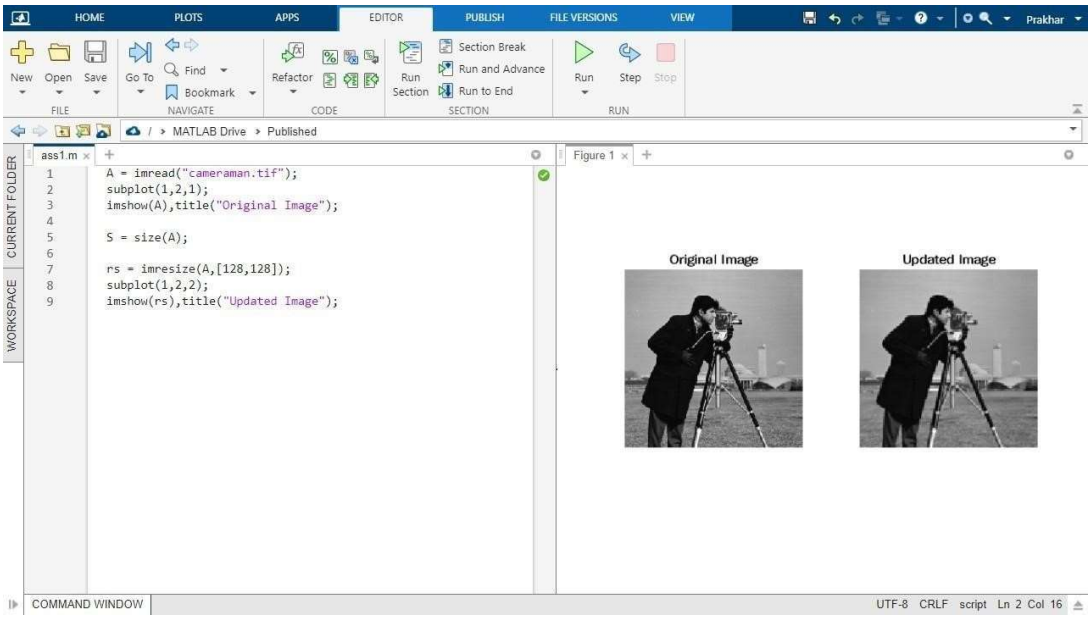


Assignment 2

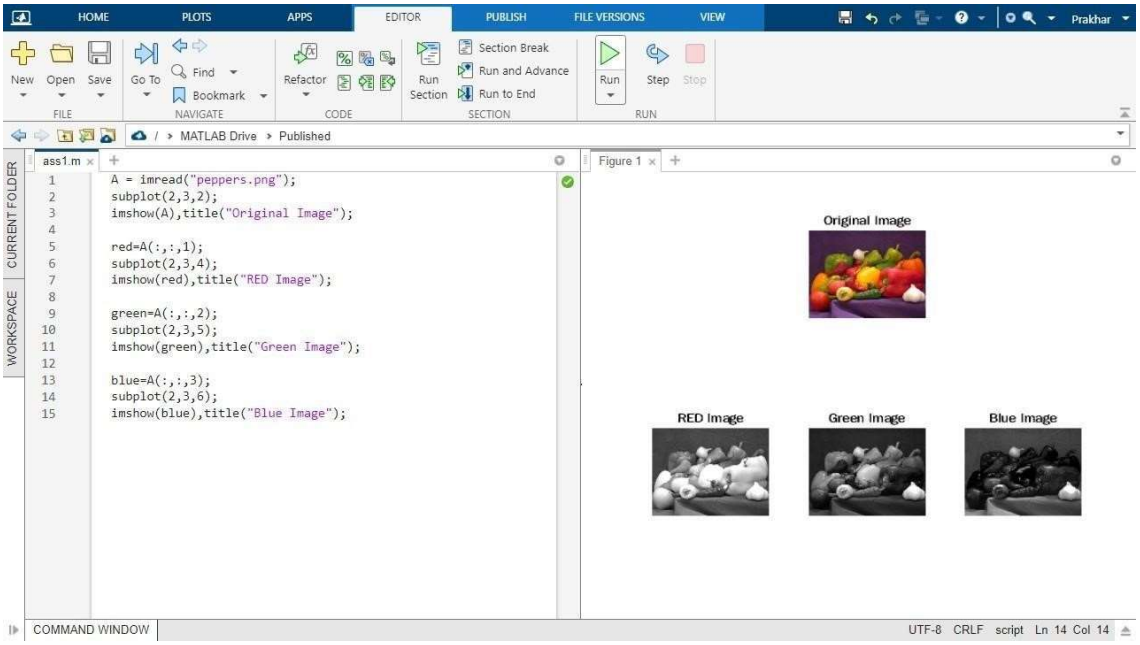
```
% Understanding Image Basic "image Resize, image type
conversion, extraction
of color band, creating a synthesic image, psedocolor
image"
```

%Reading an image as input and show the output

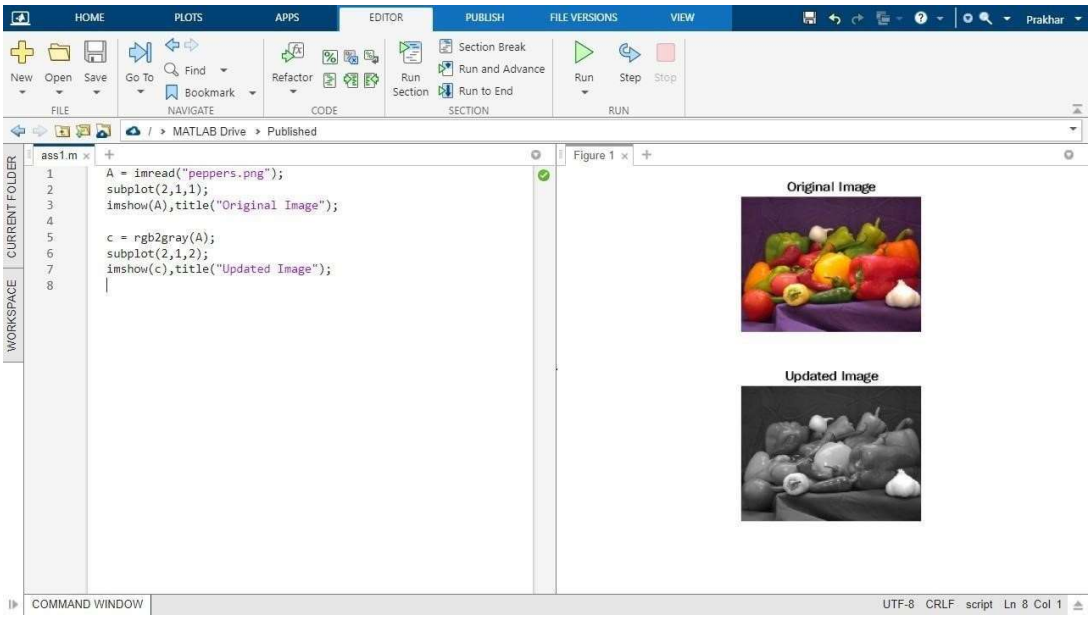
%Resize an image



%Extraction of color band of an image.



%Image type conversion



Assignment 3

% Bit Operations and Logical operations

```
a = imread('cameraman.tif');
b = im2bw(a);
c1 = imread('peppers.png');
c = rgb2gray(c1);
d = im2bw(c);
e = imresize(b,[512,1024]);
f = imresize(d,[512,1024]);
g = e | f;%bitor(e,f) function can also be used
h = imshow(g);
```

```
%c = bitcmp(b);
c = ~(b);
d = imshow(c);
```

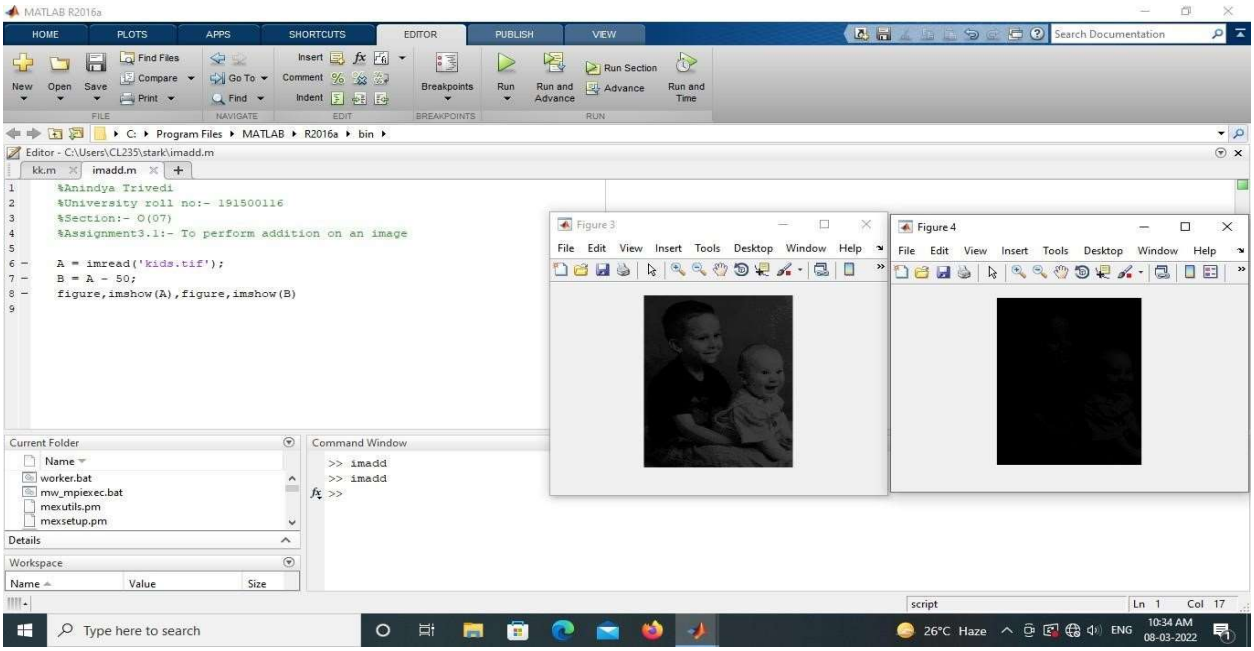
```
c1 = imread('peppers.png');
c = rgb2gray(c1);
d = im2bw(c);
e = imresize(b,[512,1024]);
f = imresize(d,[512,1024]);
g = e & f;
h = imshow(g);
```

```
d = im2bw(c);
e = imresize(b,[512,1024]);
f = imresize(d,[512,1024]);
g = ~(e & f);
h = ~(e | f);
i = (e | f) & (g);
j = ~(i);
subplot(2,2,1) , imshow(g), title('nand');
subplot(2,2,2) , imshow(h), title('nor');
subplot(2,2,3) , imshow(i), title('xor');
subplot(2,2,4) , imshow(j), title('xnor');
```

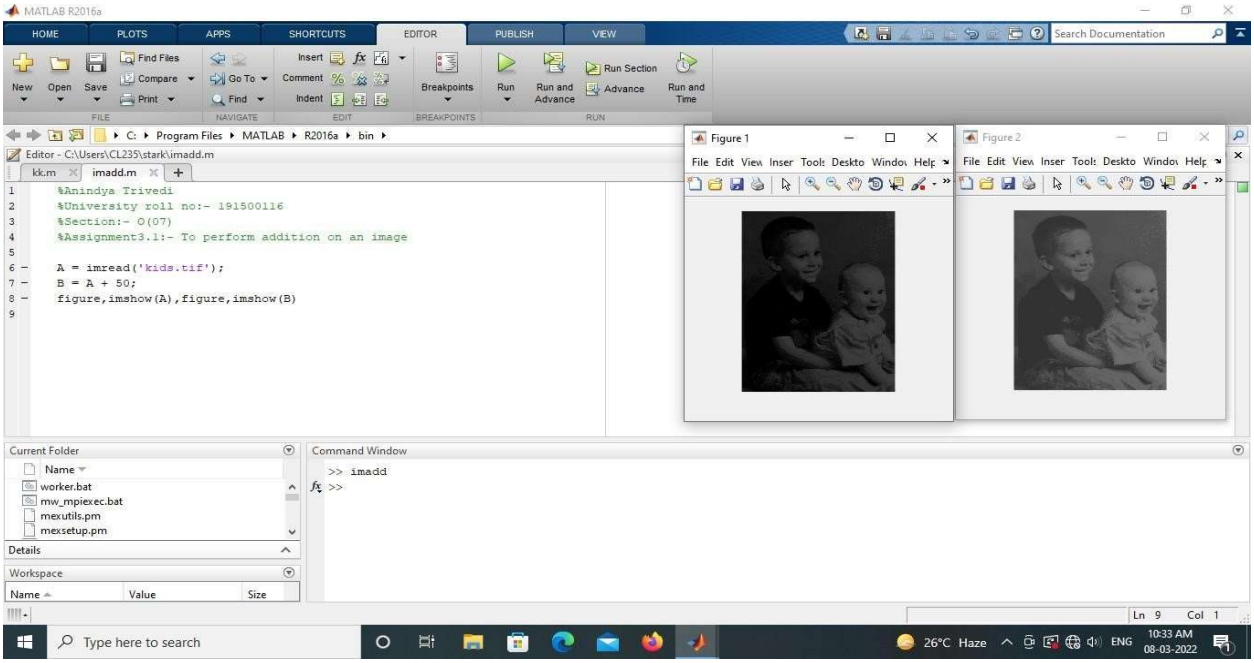
Assignment 4

Perform various arithmetic and logical operations on image.

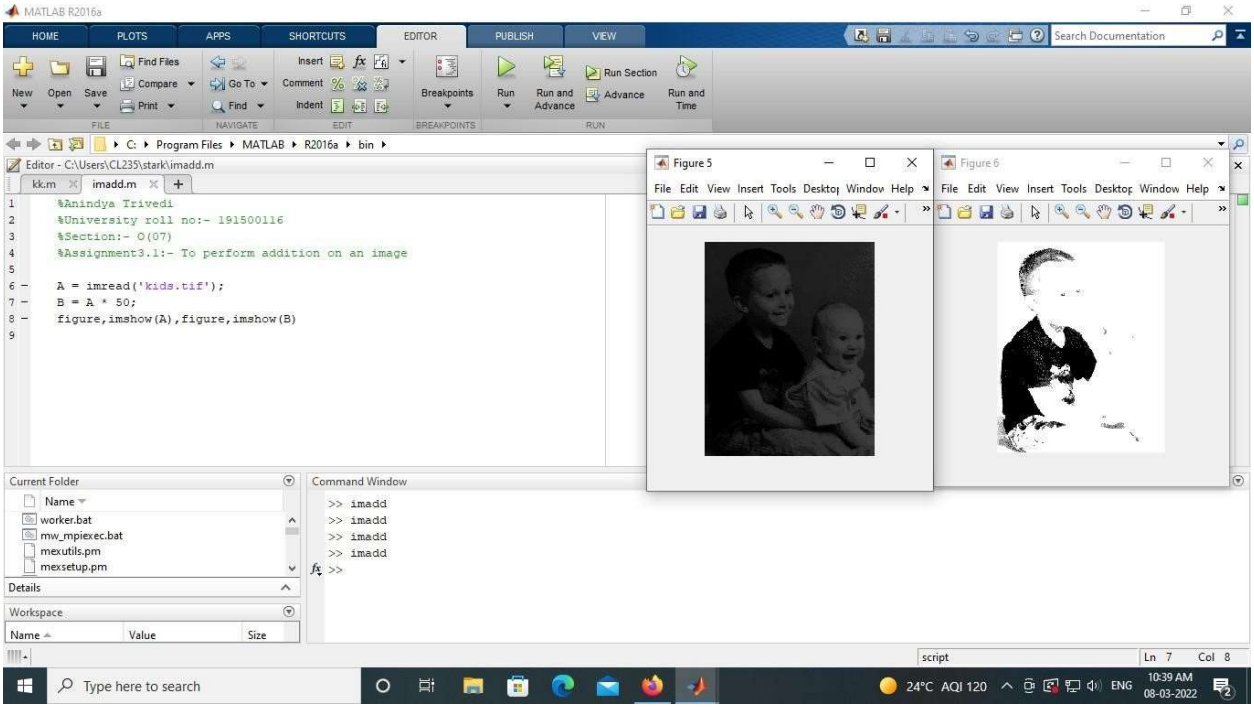
% 3.1



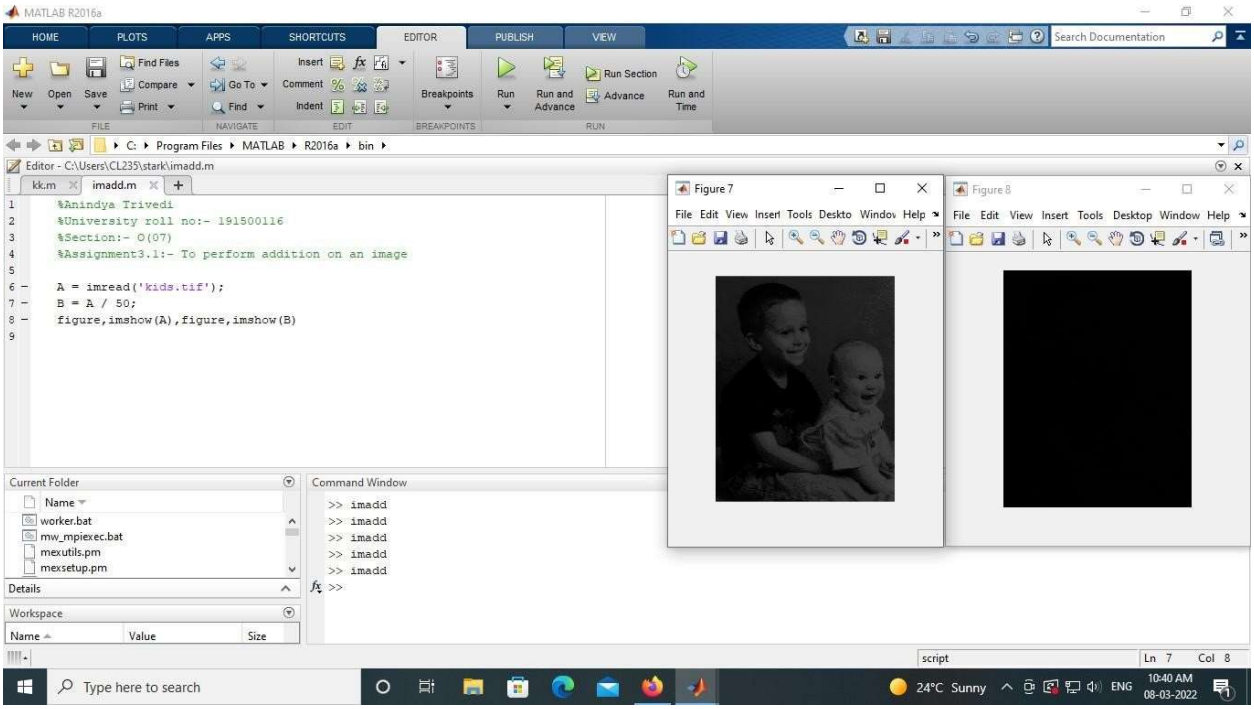
% 3.2



% 3.3


MATLAB R2016a
%Anindya Trivedi
%University roll no:- 191500116
%Section:- O(07)
%Assignment3.1:- To perform addition on an image
A = imread('kids.tif');
B = A * 50;
figure,imshow(A),figure,imshow(B)
Figure 5
Figure 6
>> imadd
>> imadd
>> imadd
>> imadd

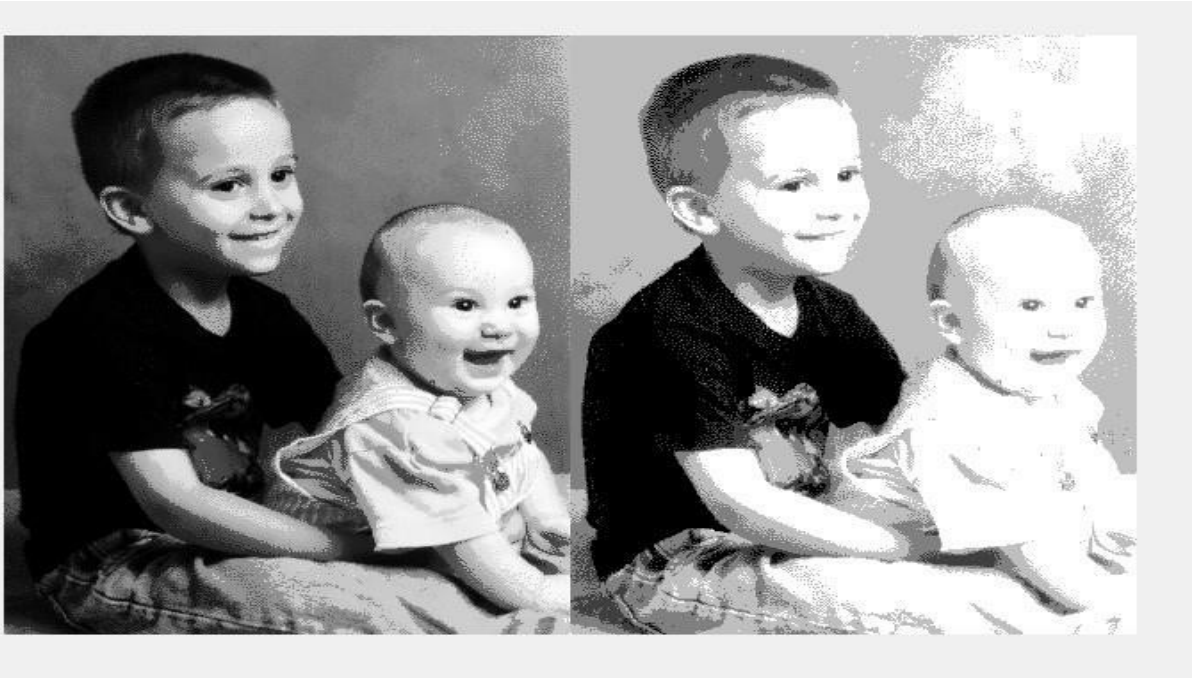
%3.4


MATLAB R2016a
%Anindya Trivedi
%University roll no:- 191500116
%Section:- O(07)
%Assignment3.1:- To perform addition on an image
A = imread('kids.tif');
B = A / 50;
figure,imshow(A),figure,imshow(B)
Figure 7
Figure 8
>> imadd
>> imadd
>> imadd
>> imadd

Assignment 5

% Perform the various image enhancement operations:
% Image negative function,
% logarithmic transformation,
% power law transformation ,
% histogramequalization contract stretching , %
plot histogram without using imhist function.

```
A=imread('kids.tif'); B=A;
[M,N]=size(B);
c=input('Enter the constant value, c='); B=double(B);
for i=1:M for j=1:N
B(i,j)=c.*log(1+B(i,j));
end end B=uint8(B);
imshowpair(A,B ,'montage');
```



Assignment 6

% Perform smoothing using linear(average filter) and order statistics
% filter(min, max & median) of varying sizes
% 5.1 -> Perform smoothing using linear and order statics filter of varying sizes

```
A = imread('peppers.png');
%imshow(A);
B = rgb2gray(A);
%imshow(B); [m,n] = size(B); B1 = double(B);
S = input('size of filter(odd number):'); f = ones(S);
C = (S+1)/2;
for i= C:m-C+1 for j=C:n-C+1
sum = 0; for k=1:S
for l=1:S
sum=sum+B1(i-C+k, j-C+l)*f(k,l);
end end
      B1(i,j)=sum/(S^2); end

end
figure(1),
subplot(1,2,1),
```

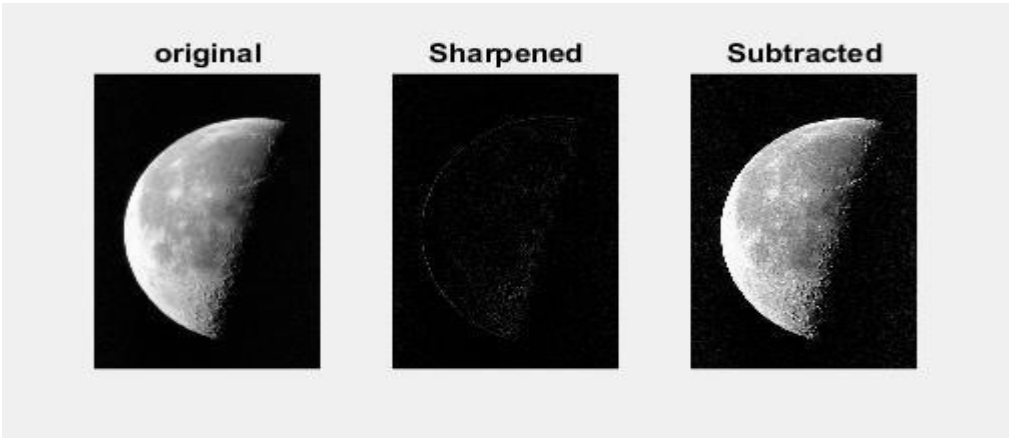

```
imshow(B),
```

```
 title 'Original',
 subplot(1,2,2),
imshow(uint8(B1)),
title 'Smooth';
```

```
>> asg5
size of filter(odd number):9
```

% 5.2 -> Sharpning an image using Laplacian filter L_4 & L_8

```
A=imread('moon.tif'); A=double(A); [m,n]=size(A);
f=[0 1 0;1 -4 1;0 1 0] s=A;
for i=2: m-1 for j=2: n-1
sum=0; for k=1:3 for
l=1:3
sum=sum+A(i-2+k,j-2+l)*f(k,l);
end end
            s(i,j)=sum;
       end
end sm=A-s;
figure(1),
subplot(1,3,1),
 imshow(uint8(A)),
 title('original'),
subplot(1,3,2),
imshow(uint8(s)),
title('Sharpened'),
subplot(1,3,3),
 imshow(uint8(sm)),
title('Subtracted')
```



Assignment 7

% 6.1 -> Find the DFT of [0 1 2 1]

```
clear all f(1:4)=[0 1 2
1]; F=zeros(1:4); for u
= 1:4 for x = 1:4
F(u)=F(u)+f(x)*(cos(2*pi*(u-1)*(x-1)/4)-sin(2*pi*(u-1)*(x-1)/4)*1i);
end end F=1/4.*F
```

```
F =

   1.0000 + 0.0000i
  -0.5000 - 0.0000i
   0.0000 + 0.0000i
  -0.5000 - 0.0000i
```

% 6.2 -> Find the DFT of [0 1 2 1] using Twiddle Matrix.

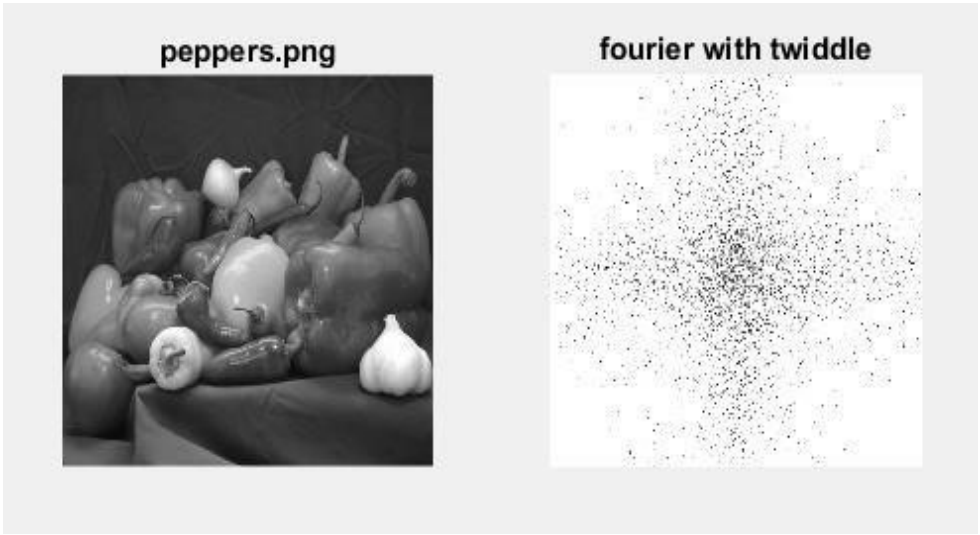
```
f=[0 1 2 1];
for u= 0:3 for
x = 0:3
val = exp(-i*2*pi*u*x/4); t(u+1, x+1) = val;
end end F = 1/4.*t*f'
```

% 6.3 -> Apply 2D Fourier Transform using 1D transformation on an actual square image.

```
F =

   1.0000 + 0.0000i
  -0.5000 - 0.0000i
   0.0000 + 0.0000i
  -0.5000 - 0.0000i
```

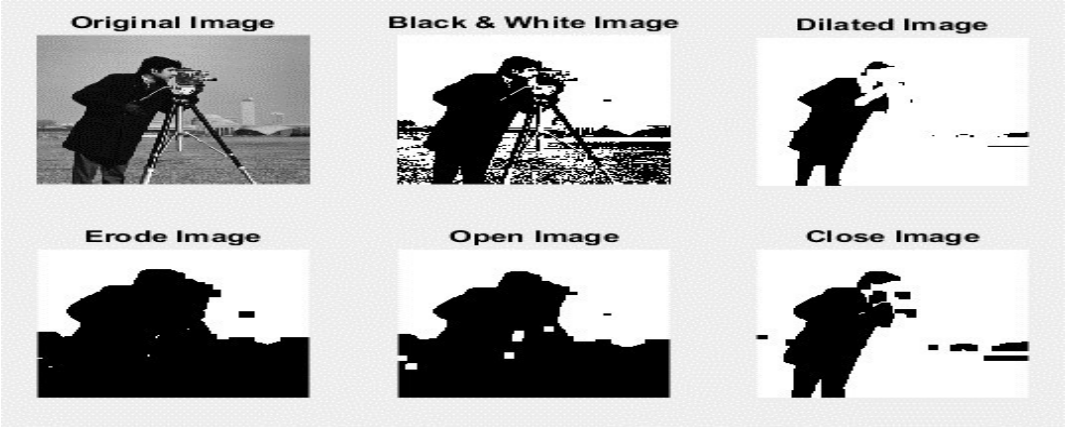
```
p=imread('peppers.png'); p=rgb2gray(p);
[m, n]=size(p);
p=imresize(p,[m, m]); p=double(p); t=zeros(m,m);
%twiddle matrix for u=0:m-1 for
x=0:m-1
aa=exp(-i*2*pi*u*x/m);
t(u+1,x+1)=aa; end end
%rowwise FR=zeros(m,m); for x=1:m
FR(x,:)=(t*p(x,:)')';
end
%coumn wise F=zeros(m,m); for y=1:m
F(:,y)=t*FR(:,y); end
F=abs(F);
figure, subplot(1,2,1), imshow(uint8(p)), title 'peppers.png', subplot(1,2,2), imshow(uint8(F)), title
'fourier with twiddle';
```



Assignment 8

% Perform various morphological operation on their applications.
% 7.1 -> Perform dilation operation
% 7.2 -> Perform Erode Operation
% 7.3 -> Perform Open Operation
% 7.4 -> Perform Close Operation

```
I=imread('cameraman.tif');
A=im2bw(I);
imshow(A);
se=strel('square',9);
dilate = imdilate(A, se);
erode = imerode(A, se);
open = imopen(A, se);
close = imclose(A, se);
figure(1),subplot(2,3,1),
imshow(I), title 'Original Image',
subplot(2,3,2), imshow(A), title 'Black
& White Image', subplot(2,3,3), imshow(dilate), title 'Dilated Image', subplot(2,3,4),
imshow(erode), title 'Erode Image',subplot(2,3,5), imshow(open), title 'Open Image',
subplot(2,3,6), imshow(close), title 'Close Image';
```



```
A=imread('cameraman.tif');
B=im2bw(A);
se=strel('square',4);
C=imerode(B,se);
D=imdilate(B,se); E = D-C;
F=bwmorph(B,'thin');
G=bwmorph(B,'thicken');
H= bwmorph(B, 'skel', 9);
%without using direct function figure(1),subplot(2,3,1), imshow(A),
title 'Original Image',subplot(2,3,2),
imshow(B),title 'Black &white Image';subplot(2,3,3),
imshow(E) ,title 'Boundary extracted Image',subplot(2,3,4),
imshow(F), title 'Thin Image', subplot(2,3,5),
imshow(G), title 'Thicken Image',subplot(2,3,6),
imshow(H), title 'Skel Image';
```

Assignment 9

% Apply 2d fourier transform using 1d transform on an actual square image.

```
clear all;
clc;
p = imread('peppers.png');
p = rgb2gray(p);
[m n] = size(p);
p = imresize(p,[m m]);
p = double(p);
t = zeros(m,m);
%twiddle matrix
for u = 0:m-1
    for x = 0: m-1
        aa = exp(-i*2*pi*u*x/m);
        t(u+1,x+1)=aa;
    end
end
%row wise
FR = zeros(m,m);
for x = 1:m
    FR(x,:) = (t*p(x,:)')';
end
%column wise
F = zeros(m,m);
for y = 1:m
    F(:,y) = t*FR(:,y);
end
F = abs(F);
P1 = abs(fft2(p));
subplot(1,3,2);
imshow(uint8(p));
subplot(1,3,2), imshow(uint8(F));
subplot(1,3,3), imshow(uint8(P1));
```